EUROPEAN PATENT SPECIFICATION

(54) Method of software protection.

(73) Proprietor: **International Business Machines
Corporation
Old Orchard Road
Armonk, N.Y. 10504(US)**

(72) Inventor: **Matyas, Stephen Michael
R.D. 5 Box 19F
Kingston New York 12401(US)**
Inventor: **Oseas, Jonathan
Box 147
Hurley New York 12401(US)**

(74) Representative: **Bailey, Geoffrey Alan
IBM United Kingdom Limited Intellectual
Property Department Hursley Park
Winchester Hampshire SO21 2JN(GB)**

Description

This invention is directed to methods of software protection and, more particularly to a cryptographic method for discouraging the copying and sharing of purchased software programs by allowing an encrypted
5 program to be run on only a designated computer or, alternatively, to be run on any computer but only by the user possessing a designated smart card.

With the proliferation of so-called micro computers or personal computers, there has been an explosion in the writing and publishing of software for these computers. The investment in time and capital in the development of a good software program can be substantial, and in order to recoup this investment, the
10 authors, copyright owners and/or publishers must rely on royalties and the amortization of costs produced by the sale of the programs. Since programs are generally distributed on inexpensive floppy diskettes, the end user often does not appreciate the substantial costs in the production of the programs. Moreover, the disk operating systems of most micro computers feature a disk copy utility which enables the end user to easily make back up copies of program diskettes. The result has been misuse of the utility to make
15 unauthorised copies. For example, two or more potential end users desiring a program may pool their resources and buy one copy of the program and then duplicate both the program diskette and the copyrighted manual that accompanies the program diskette. As another example, a small business having several micro computers may buy a single copy of a program and then duplicate the program diskette and manual in order to distribute copies to each micro computer station in the company. Both of these
20 examples are clear violations of the copyright laws, but searching out the prosecuting violators is often impossible. The net result is a substantial loss of revenues to software authors and publishers. These revenues are needed in order to finance the development of new and improved software programs as well as to provide a reasonable profit to those who produced the programs that are copied without authorisation.

The problem of unauthorised copying and use of programs has been addressed by the prior art. US-A-
25 4,120,030 to Johnstone discloses a computer software security system wherein the data address portions of a set computer instructions are scrambled in accordance with a predetermined cipher key before the instructions are loaded into an instruction memory. The data involved in the program is loaded into a separate data memory at the addresses specified in the original, unscrambled program. An unscrambler circuit, which operates in accordance with the cipher key, is coupled in series with the data memory
30 address input conductors.

US-A-4,168,396 to Best discloses a microprocessor for executing computer programs which have been enciphered during manufacture to deter the execution of the programs in unauthorised computers. US-A-4,278,837 to Best discloses a crypto-microprocessor chip that uses a unique cipher key or tables for deciphering a program so that a program that can be executed in one such chip cannot be run in any other
35 microprocessor. US-A-4,433,207 to Best discloses an integrated circuit decoder for providing micro computer users with access to several proprietary programs that have been distributed to users in cipher. The decoder chip can decipher a program if an enciphered key called a "permit code" is presented to the decoder chip.

US-A-4,446,519 to Thomas discloses a method for providing security for computer software by
40 providing each purchaser of a software package with an electronic security device which must be operatively connected to the purchaser's computer. The software sends coded interrogation signals to the electronic security device which processes the interrogation signals and transmits coded response signals to the software. The programs will not be executed unless the software recognizes the response signals according to preselected security criteria.

45 The various schemes disclosed by these patents require specialized and dedicated hardware for accomplishing the security feature. Generally, these schemes are cumbersome and expensive to implement and therefore not commercially acceptable. What is needed is a software protection scheme which is simple and inexpensive to implement, avoiding, in its simpler forms, the need for specialised and dedicated hardware, but still able to be extended by the use of additional hardware and which is attractive to a large
50 number of diverse software publishing houses.

Further prior art methods for preventing unauthorized copying are disclosed in EP-121853-A, EP-089876-A, and the article "Combatting software piracy by encryption and key management", Computer, Vol. 17, No. 4, April 1984, pages 68-73, Long Beach, California, US; D J Albert et al.

Therefore, according to the present invention, there is provided, as the core method, a method of
55 software protection comprising the steps of encrypting a program with a file key, and writing the encrypted program on a storage medium as a program file; providing a computer, having a protected memory and a cryptographic facility, with a secret cryptographic key identifier cryptographically derived from a number identifying the computer; providing a user of the storage medium containing the encrypted program with a

password cryptographically derived from at least the file key, the number identifying the computer, a number identifying the storage medium, and a number identifying the program; entering the password into the computer when the medium is loaded thereinto; deriving the file key from at least the password, the storage medium number, the program number, and the cryptographic key identifier; and decrypting into
5   executable form at least a portion of the program using the derived file key.

Such a software protection scheme is inexpensive to implement and is essentially transparent to the end user so that it does not detract from the commercial appeal of the software program. It can be arranged to allow copying of the program diskettes to a hard disk or for purposes of making backup copies yet limit the simultaneous beneficial use of multiple copies to one or more designated computers or, alternatively,
10   limit the use to one unique, non-reproducible and portable device or, alternatively, to be run on any computer but only by the user possessing a designated smart card.

Put another way, ideally in this arrangement, each program offering sold by a software vendor is encrypted with a unique file key and then written on the diskette. A user who purchases a diskette containing an encrypted program must first obtain a secret password from the software vendor. This
15   password will allow the encrypted program to be recovered at a prescribed, designated computer having a properly implemented and initialised encryption feature which may be stored in Read Only Memory (ROM), for example. As part of an initialisation process, when the program is first loaded, it polls the user to input the password. The password is written by the program in the header record of the file, and once written in the header record, the program will not prompt the user to input his or her password on subsequent uses of
20   the program. When the diskette is loaded at the proper computer, the encrypted program or a controlling portion of it is automatically decrypted and written into a protected memory from which it can only be executed and not accessed for non-execution purposes. In an alternative embodiment, the user is not confined to a prescribed, designated computer but may use the program on other, different computers provided that such computers incorporate a smart card. A smart card as defined herein is one having a
25   crypto capability, typically implemented by incorporating a micro-circuit on the card. The smart card is issued to the user when the user purchases a computer. The smart card is preinitialised by the computer manufacturer with a secret parameter unique to that card. The procedure is similar except that in this case, the password used in conjunction with the smart card allows the user to decrypt and execute the program on any computer having a properly implemented and initialised encryption feature. The smart card
30   embodiment can be further modified to allow portability with a combination of both a Public Key (PK) algorithm and the Data Encryption Standard (DES) algorithm. In this case, the designated public registry (key distribution centre) additionally personalises the card with the public key of the computer manufacturer as well as a unique secret card key (the DES key). When a customer buys software, the customer automatically obtains a personalised intelligent secure card (smart card). Each different program recorded
35   on the diskette is encrypted under a different file key designated by the supplier of the software. The customer then obtains an authorisation number and password from the software vendor, as before. The password is written in the header of the file on the diskette. The computer is also personalised with a unique key pair, a public computer key and a secret computer key. However, the public computer key is first decrypted under the secret key of the computer manufacturer and stored in the computer in that form.
40   When the program diskette is used, there is a handshake protocol between the smart card and the computer which, in effect, recovers the file key and enciphers it under the public key of the computer. This protocol is such that the handshake will work only at a suitable computer with a public key algorithm and a properly installed key pair; i.e, a secret computer key and a public computer key decrypted under the secret key of the designated public registry. The advantage to this approach is that the file key can be
45   encrypted using a public key, and it is not necessary for a universal secret DES key to be stored on each smart card. The protocol for password generation and distribution is unaffected by the modification to the internal protocol; i.e., it is the same for the DES only smart card as for the DES/PK smart card.

The present invention will be described further, by way of example, with reference to embodiments thereof, as illustrated in the accompanying drawings, in which:-
50   Figure 1 is a block diagram of an overview of one form of system according to the invention, which does not involve the use of a smart card;
Figure 2 is an illustration of a program diskette format compatible with the system of Figure 1;
Figure 3 is a flow diagram illustrating the password generation key management associated with the system of Figure 1;
55   Figure 4 is a flow diagram illustrating the operation of the computer cryptographic facility key management associated with the system of Figure 1;
Figure 5 is a block diagram of an overview of another form of system according to the invention which uses a smart card;

Figure 6 is a flow diagram illustrating the password generation key management associated with the system of Figure 5;

Figure 7 is an illustration of a program diskette format compatible with the system of Figure 5;

Figure 8 is a flow diagram illustrating the operation of the computer cryptographic facility key management associated with the system of Figure 5 and using the DES algorithm;

Figure 9 is a flow diagram illustrating the operation of the smart card using the DES algorithm;

Figure 10 is a flow diagram illustrating the generation of program diskettes by a software vendor;

Figure 11 is a block diagram of the components internal to a computer which incorporates the cryptographic facility required in the practice of the present invention;

Figure 12 is a block diagram showing the basic components of a smart card usable in some embodiments of the invention;

Figure 13 is a flow diagram illustrating the operation of the computer cryptographic facility key management with the smart card using the DES/PK algorithm;

Figure 14 is a flow diagram illustrating the operation of the smart card using the DES/PK algorithm;

Figure 15 is a flow diagram illustrating a first operation of the cryptographic facility which derives a file key from user input parameters;

Figure 16 is a flow diagram illustrating a second operation of the cryptographic facility which causes a random number to be generated;

Figure 17 is a flow diagram illustrating a third operation of the cryptographic facility which uses a password from a smart card and a random number to derive a file key to decrypt a program;

Figure 18 is a flow diagram illustrating a fourth operation, similar to the first, of the cryptographic facility for generating a file key using a different algorithmic procedure than used in the first operation;

Figure 19 is a flow diagram illustrating a fifth operation, similar to the second, of the cryptographic facility for generating a file key using the same algorithmic procedure as used in the forth operation;

Figure 20 is a flow diagram illustrating a sixth operation of the cryptographic facility for generating a file key from user input parameters for encrypting data; and

Figure 21 is a flow diagram illustrating a seventh operation of the cryptographic facility which accepts a password from a smart card for generating a file key for encrypting data.

In the description which follows, the notations "e" and "d" are used to denote encrypted and decrypted, respectively. For example, "ePKt(KF)" means the key KF is encrypted under the key PKt. Similarly, "dSKu(PKt)" means the key PKt is decrypted under the key SKu. Also, the term "personal computer" is intended to cover so-called "smart terminals" which may include personal computers connected to a main frame computer in the network.

Referring now to the drawings and more particularly to Figure 1, there is shown an overview of the first embodiment of the invention. The system according to this embodiment comprises a personal computer 10 having a crypto protected storage facility 101. Each program offering 12 sold by the software vendor is encrypted with a unique file key, KF, and then written on the diskette. The format of the diskette is generally shown in Figure 2. One type of cryptographic method that may be used is a cipher block chaining technique which requires an initialising vector. If desired, different initialising vectors using the same file key, KF, can be used to encrypt the same program written on different diskettes. This ensures that different cipher text is produced for each diskette and prevents differences in the plain text from being observed by comparing the corresponding cipher text. The initialising vector is written in the header record on the diskette.

A user, who purchases a diskette containing an encrypted program, must first obtain a secret password from the software vendor. This password will allow the encrypted program to be recovered at the prescribed, designated personal computer 10 having a properly implemented and initialised encryption feature. The secret password allows the particular program to be decrypted, and thus executed, only on a particular personal computer. This secret password is unique to the particular program and computer where it is to be recovered and executed. The password will not allow other encrypted programs to be recovered on that computer, nor will it allow that same encrypted program to be recovered on a different computer. Optionally, the user could be given a second password that would allow the encrypted program to be recovered on a designated backup computer. Except for the backup computer, the user would ordinarily be expected to pay an extra fee for each additional password that would allow the encrypted program to be recovered on a different computer.

Each diskette has a unique serial number written on the diskette envelope or outer cover (not shown) and visible to the user. As shown in Figure 2, this serial number is also recorded in the header record of the diskette. Also recorded in the header of the diskette is the program number. In the specific example shown in the figure, the program number is "12" and the serial number or diskette number is "3456". A multi-digit

authorisation number is obtained by encrypting the program number and diskette serial number, concatenated together, under a secret cryptographic key available to and known only by the software vendor. An n-bit portion of the authorisation number is also written on the diskette envelope except that it is covered by a thin metallic film much like that used by the instant lotteries to hide numbers on lottery cards. Where n is
5　equal to 16, for example, there are 65,536 possible numbers for the portion of the authorisation number written on the diskette envelope and therefore, one would have only a chance of one in 65,536 of accidently guessing a correct number written on the diskette envelope.

When a password is requested, an authorisation number of reference is generated in the same manner as that for generating the authorisation number. For each password initially issued, a record is made in a
10　data base that this is a first use of the authorisation number of reference in the process of issuing the password to a requesting user, and a record is also made of the password which is issued. Therefore, for each request of a password, a first use check is made to determine whether the authorisation number of reference has been previously used for generating the password.

The procedure is illustrated in Figure 3. After purchasing a diskette, the user places a telephone call to
15　the software vendor. It is assumed that the user will not accept a diskette whose authorisation number has been exposed, i.e. where the metallic film has been scratched off. The customer provides the software vendor with the program number, the n-bit portion of the authorisation number, the diskette serial number, and the computer number. Each computer 10 has a unique identification or number that is provided on the cover, for example, by a press-on label visible to the user. This identification or number is associated with
20　the secret key of the crypto facility of the computer. The program number and the diskette serial number are loaded into register 20. Recall that for the specific example shown in Figure 2, these numbers are "12" and "3456", respectively. Then in block 21, the software vendor simply encrypts the provided program number and diskette serial number, concatenated together, with a special secret key, SK, used only to generate multi-digit authorisation numbers. The n-bit portion of the authorisation numbers written on the
25　diskettes are produced with the same encryption technique. The multi-digit authorisation number or reference is first checked in the software vendor's database to determine if the authorisation number has been used before. Alternatively, the software vendor could perform this check using the program number and diskette serial number. In this case, the software vendor simply records the program number and diskette serial number in his data base whenever a password has been issued for those numbers. If the
30　authorisation number or the program number and diskette serial number have been used before, the software vendor knows that a password has been issued for that program number and diskette serial number and that this password has been recorded in his data base. In that case, the password is retrieved and reissued to the caller. This process is represented by block 22 in Figure 3. On the other hand, if the result of this process indicates a first use, then in block 24 the designated n-bit portion of the authorisation
35　number of reference produced in block 21 is compared with the n-bit portion of the authorisation number provided by the caller. If a match is obtained, the software vendor generates a special password that will allow the encrypted program to be decrypted and executed at the designated computer. To accomplish this, the software vendor forwards an electronic message to the key distribution centre 14, passing the program number, diskette serial number and computer number. The key distribution centre 14 encrypts the
40　computer number with a key, KT, in encryption block 26 to produce an encryption key,eKT(TR), unique to that particular computer. Alternatively, the key eKT(TR) could be obtained from a table of stored keys. The program number and the diskette serial number in register 28 are then encrypted in encryption block 30 with the key eKT(TR) to produce a cryptographic key unique to the program and computer. In the example illustrated, this cryptographic key is eKT(PG123456). The key distribution centre 14 then returns the
45　cryptographic key to the software vendor. To further enhance the security of the system, encryption can be used between the software vendor and the key distribution centre to protect the secrecy of the cryptographic key.

Meanwhile, the software vendor has obtained from its database the file key, KF, corresponding to the program number provided by the caller as indicated by table 32.
50　The key KF is then encrypted in encryption block 34 with the cryptographic key returned by the key distribution centre 14 to generate the requested password. The generated password is then given to the caller. Passwords may be, for example, 64-bits long and therefore cannot be guessed or derived from other information available to the caller. As a last step, the software vendor now makes a record in his data base that this is the first use of the authorisation number of reference in the process of issuing the password to a
55　requesting user, and he also records the calculated password in his data base.

As part of the initialisation process, when the program is loaded in the computer 10, it polls the user to input the password. The password is written by the program in the header record of the file as shown in Figure 2, and once written, the program will not prompt the user to input his or her password on subsequent

uses of that program. For example, a protocol could operate such that the computer always reads the header of the diskette looking for a recorded password. If no password is found, it prompts the user to enter the password and then writes the password in the header. If the password is found in the header, the computer uses this password in lieu of prompting the user to enter a password. The user could also be
5   provided with an override to enter the password in case the password recorded in the header fails to produce the correct file key, KF; i.e., the encrypted file is not recovered properly with the recovered key KF. Note that from the software vendor's viewpoint the password need not be kept secret since it does not unlock other encrypted programs.

     Optionally, the procedure for issuing passwords at the software vendor could be fully automated by
10 using a voice answer back system in conjunction with a multi-frequency tone input. For example, the caller would be prompted to enter the appropriate numbers using the multi-frequency tone keyboard on a telephone, and these numbers would be repeated to the user for verification. If a proper authorisation number is given, an electronic message is sent to the key distribution centre 14 to obtain the necessary cryptographic key. This message is used to calculate the password which, in turn, is repeated to the caller
15 using the automated voice system. The process of obtaining the password from the software vendor could be automated still further by initiating a communications session between an initialisation program in the personal computer 10 and a password distribution program located in the computer system of the software vendor. In this case, the user would call the 800-number and initiate the session. The program number and diskette serial number could be read from the header record of the diskette where they have been written
20 by the software vendor. The computer number could be stored within the system and provided automatically also. The user would be prompted at the appropriate point in the session to enter the authorisation number through the keyboard. The obtained password would be written automatically to the header record of the diskette file.

     If necessary, the user can contact the software vendor at any later time to re-receive his or her
25 password. To do this, he supplies only the program number and diskette serial number, which is enough information to allow the software vendor to determine that a password has already been issued for that pair of numbers and to recover the password value which has been recorded previously in his data base. In essence, the caller is given an already calculated password for any program number and diskette serial number in the vendor's database. This option does not weaken the system but merely makes it more
30 usable by end users.

     When the diskette is loaded at the proper computer 10, the encrypted program or a portion of it is automatically decrypted and written into a protected memory 101 from which it can only be executed and not accessed for non-execution purposes. This is shown in Figure 4 where the cryptographic facility 101 of computer 10 reads password, program number and diskette serial number from the file header. The
35 program number and diskette serial number concatenated together are encrypted in encryption block 103 with the encryption key for that particular computer to produce a decryption key which is used in decryption block 105 to decrypt the password and produce the secret file key, KF, that is used in decrypting the program or a portion of the program. The procedure used is the Data Encryption Standard (DES). Note that only the designated computer is capable of generating the decryption key what will produce the file key,
40 KF.

     Turning now to Figure 5, there is shown an alternative embodiment which is an extension of the first to allow portability of an encrypted program by means of a smart card 16. In this embodiment, when the user purchases a computer, the user is also issued with a smart card 16 that is pre-initialised by the computer manufacturer with a secret parameter unique to that card. A user, who purchases a diskette containing an
45 encrypted program, also obtains a secret password from the software vendor as before, except here, the password is used in conjunction with the smart card to allow the user to decrypt and execute the program on any personal computer having a properly implemented and initialised encryption feature.

     The process is illustrated in Figure 6. To obtain the secret password, in this case, the user provides the smart card number rather than the computer number. Each smart card has a unique identification or card
50 number that can be read by the user. If the authorisation number is valid and another request has not been made for that program number and diskette serial number as indicated by the compare blocks 22 and 24, the software vendor generates a special password that will allow the encrypted program to be decrypted at any personal computer with a valid encryption feature when used with that smart card. Again, the software vendor obtains a unique cryptographic key from the key distribution centre 14 which is used in conjunction
55 with the secret file key to generate the requested password. In this case, the key distribution centre 14 is owned, controlled or established under the direction of the computer manufacturer and uses the card number to obtain a corresponding card encryption key, KP, from table 27. Alternatively, the key KP can be generated from a secret key belonging to the key distribution centre in a similar manner as the eKT(TR)

keys are generated using secret key KT as shown in Figure 3. This encryption key is then used to encrypt the program number and diskette serial number in encryption block 30 to produce the cryptographic key that is returned to the software vendor. As before, the security of the system can be enhanced by using encryption between the key distribution centre and the software vendor to protect the secrecy of the communicated keys. Otherwise, the protocol is the same as that for the first embodiment. The password is written in the header record of the diskette as illustrated in Figure 7. The same options are available for automating the process at the software vendor. Likewise, an initialisation program in the personal computer 10 can be used to automatically obtain the password from the software vendor, except here the smart card number must be entered instead of the computer number. The smart card number could be read from the smart card or from a location in the computer where it had been previously stored.

When the diskette is loaded at any authorised computer and the smart card 16 has been inserted into a proper reader device allowing the card and computer to carry on an electronic dialogue, the encrypted program is automatically decrypted and written into the protected memory 101 from which it can only be executed. More specifically, as shown in Figures 8 and 9, the computer 10 reads the program number, diskette serial number and password from the file header and passes these with its computer number to the smart card 16. In the smart card 16, the computer number is encrypted in encryption block 161 with a universal key, KT stored on every smart card. The program number and diskette serial number are encrypted in encryption block 162 with a key KP which is unique to and stored in the smart card. The output of encryption block 162 is a decryption key that is used by decryption block 163 to decrypt the password to produce the secret file key, KF.

Meanwhile, the cryptographic facility 101 of computer 10 produces a random number T using a random number generator 107 or the system clock. As part of the internal protocol interchange between the computer 10 and the smart card 16, the random number T is passed to the smart card where it is exclusive ORed with the file key, KF. The resulting output is encrypted in encryption block 164 with the output of encryption block 161 to produce a computer password that is then passed back to the computer. This computer password is then decrypted in decryption block 109 using a key unique to the computer. The output of decryption block 109 is exclusive ORed with the random number T to produce the secret file key, KF. Note that passwords generated by the card are time variant. If intercepted and replayed back into the computer at a later time, they will not allow another copy of the encrypted program on a different diskette to be decrypted and executed.

Figure 10 shows the process of generation of program diskettes. First, a clear diskette 36 containing the program is supplied by the software vendor to the software distributer. The software distributer then encrypts the program in encryption block 38 using a key KF to produce and encrypted diskette 40. The key KF could be common to a program or unique for each diskette. The encrypted diskette is then copied with a disk copier 42 to produce encrypted diskettes 44 for sale to users. Obviously, in the process illustrated, the software distributer and the software vendor could be one and the same.

Figure 11 shows the cryptographic facility 101 in the personal computer 10. The cryptographic facility is a secure implementation containing the Data Encryption Standard (DES) algorithm and storage for a small number of secret keys. It can be accessed logically only through inviolate interfaces secure against intrusion, circumvention and deception which allows processing requests via a control line, key and data parameters to be presented, and transformed output to be received. The cryptographic facility comprises a ROM, such as an EAPROM (Electronically Alterable and Programmable Read Only Memory), 113 that contains the computer key. Additional ROM 114 contains programs for key management sequence generators and disk loader. Additional Random Access Memory (RAM) 115 contains a parameter output buffer, a parameter input buffer, intermediate storage for parameters, data and keys, and additional storage for the decrypted programs. The RAM 115 is the protected memory of the cryptographic facility 101, and decrypted programs stored here can only be executed and not accessed for non-execution purposes.

Figure 12 shows the basic components of a smart card used in certain embodiments of the invention. The smart card must contain a microprocessor chip 165 for executing the crypto algorithm. Further, the card is provided with memory for storing the key KP and card number. The key KP, which is unique to the card and kept secret, is stored in private memory 166, while the card number is stored in public memory 167. Power for the microprocessor chip and supporting memories is derived from the computer 10.

The described protocol allows other software vendors to market encrypted programs that will operate with a personal computer or with a personal computer and smart card with no loss of security to protected software or the secret keys or parameters that support the system. To interface to the system, it is only necessary for the software vendor to forward an electronic message to the key distribution centre, passing the program number and computer number or, in the case of an implementation supporting smart cards, passing the program number and smart card number. The key distribution centre returns a cryptographic

key unique to the program number and computer number or smart card number which the software vendor used in conjunction with the secret file key under which the program has been encrypted to generate the required password. To ensure that the same key is not produced by two different software vendors who have two different programs with identical program numbers, the protocol can be modified slightly by assigning a unique two or three digit code number to each software vendor and then merely redefining the program number to consist of the software vendor code number followed by the software vendor defined program number, i.e. by as many digits as are necessary to distinguish different programs offered by that software vendor. Thus, the program number and diskette serial number used in calculations is replaced by vendor number, program number and diskette serial number.

Referring now to Figures 13 and 14, as already alluded to, the second embodiment of the invention using the smart card can be extended using a public key algorithm. More specifically, a public key (PK) algorithm is also installed in the smart card and computer in addition to the DES algorithm. The advantage of doing this is that the protocols for the three techniques are very similar. Introduction of the PK algorithm does not affect the password generation/distribution process described with reference to Figure 6. In this embodiment, the main advantage of the PK algorithm is achieved, namely that a universal secret key need not be stored on the smart card. The card manufacturer personalises the card with the key KP. The manufacturer also personalises the card with PKu, the public key of the registry. When a customer buys software, s/he automatically gets a personalised smart card. Each different program recorded on a diskette is encrypted under a different file key, KF, designated by the supplier of the software. When a customer buys a program on a diskette, the customer mails a proof of purchase coupon to the vendor along with his or her name and the serial number from his or her smart card. In lieu of this, the proof of purchase coupon contains an authorisation number that must be scratched clear as previously described. The numbers are sparse such that no one could easily guess a number that represents a valid authorisation number. The customer calls the vendor and asks for a special password to activate his or her diskette. This number is given out only after the proof of purchase authorisation number has been supplied and checked against an active online file to make sure that no one has already used the number. If all conditions are satisfied, the vendor asks for the customer's card number and uses this to access another active online file to obtain the key, KP, associated with the smart card as described with reference to Figure 6. The vendor then enciphers the file key, KF, of the purchased program and gives this number to the customer as a password. The customer then goes to his or her computer and causes the password to be written in the header of the file on the diskette as previously described. As shown in Figure 20, the header of the diskette has stored therein the password eKP(PG123456(KF) where KF is the secret file key of the program. Thus, to this point the procedure is similar to that described with respect to the smart card/DES only algorithm.

The computer manufacturer personalises the computer with a unique key pair, the computer public key, PKt, and the computer secret key, SKt. The computer manufacturer has the public key of the computer recorded in a public registry. In effect, this means that PKt is stored in the form dSu(PKt), where SKu is the secret key of the registry and PKt is the public key of the computer. This value dSu (PKt) is also stored in the computer.

To use a diskette at any computer, there is a handshake protocol used between the smart card 16 and the computer 10 which, in effect, recovers the file key, KF, from encipherment under the key KP and enciphers it under the public key, PKt, of the computer. More specifically, the public key, PKt, of the computer 10 in the form dSu(PKt) is encrypted in encryption block 161 with the public key of the registry, PKu, to produce the public key, PKt, of the computer. The notation dSu (PKt,0) means that the public key PKt with several redundancy bits (0 bits in this case) concatenated with it is decrypted under the secret key SKu. SKu is the secret key belonging to the computer manufacturer. The redundancy bits are added to the message so that upon decryption, one can ensure that no spurious text is decrypted and used as the key PKt. Ordinarily, 16 to 64 bits of redundancy is enough. As a result of these redundancy bits, the output of encryption block 161 must be checked to ensure that the redundancy bits compare with a prestored constant value. If they do, then one can be certain that the recovered PKt is the public key of some computer manufactured by the computer manufacturer. This ensures that an opponent can not get a smart card to use a public key PKt except one issued by the computer manufacturer. Of course, it will be recognised by those skilled in the art that the block with the redundancy bits will typically be longer that the public key PKu which will necessitate splitting the block and performing the encryption process under PKu using chaining techniques well known in the art.

The program number read from the diskette is also encrypted with the key KP in encryption block 162 to generate the key eKP(PG123456) which is used to decrypt the encrypted key KF in decryption block 163. The output of decryption block 163 is exclusive ORed with a random number T produced by random number generator 107, and the result is encrypted in encryption block 164 with the key PKt to produce the

8

secret file key, KF, exclusive ORed with the random number T encrypted under the public key of the computer, PKt. This password is passed by the smart card 16 to the cryptographic facility 101 of the computer 10 where it is decrypted in decryption block 105 using the secret key, SKt, of the computer and then exclusive ORed with the random number T. The file key, KF, is now in a form that can be used at the
5   computer. The protocol is such that the handshake will work only at a suitable computer with a public key that has been properly recorded in the registry, i.e. for which PKt has been deciphered under the secret key of the registry.

The advantages of the mixed public key and DES embodiment are several. No secret universal key needs to be stored on the card since all universal keys used in the system are public keys. Even if the file
10   key, KF, is discovered, there is no way for an adversary to cause a clear key to be accepted by the card. The value T sent from the computer prevents an adversary from tapping the interface to obtain the encrypted key KF and replaying it into a computer. If it were easy to input parameters across the interface from a dummy card, such an attack is thwarted by incorporating the value T.

The described protocol allows software vendors to market encrypted programs that will operate with a
15   particular computer and smart card with no loss in security to the protected software of a given vendor or the secret keys or parameters that support the system. To interface to the system, it is only necessary for the software vendor to forward an electronic message to the key distribution centre, passing the program number and computer number or, in the case of an implementation supporting smart cards, passing the program number and smart card number. The key distribution centre returns a cryptographic key unique to
20   the program number and computer number or smart card number. The software vendor uses this cryptographic key in conjunction with the secret file key under which the program has been encrypted to generate the required password.

The cryptographic facility 101 supports a limited set of cryptographic operations for key management purposes. These operations are controlled by seven microcode routines stored in ROM and initiated by
25   decoding of an operation code corresponding to a specific operation. The first of the seven operations is illustrated in Figure 15 of the drawings. With no smart card, the first operation is decoded in operation control unit 102, and the address for the microcode is stored in register 104. The first operation accepts a password P1 and a number P2 representing the concatenation of the program number and diskette serial number read from a file header record, and from these input parameters, it derives a file key, KF, that is
30   used only by the cryptographic facility to decrypt and encrypted program for the sole purpose of executing the program. More specifically, the program number and diskette serial number, concatenated together, are encrypted under a "burned in" key eKT(TR5678) in encryption block 103 to produce a cipher text output C1. Then, the password is decrypted in block 105 using as a key the cipher text C1 produced by block 103 to produce cipher text C2 representing the file key, KF. The user has no access to the file key; i.e, it is kept
35   secret.

With the smart card, the second operation shown in Figure 16 causes a random number T to be generated inside the cryptographic facility. A special latch 106 in the cryptographic facility is also set on as a result of this operation. The value of T is generated by the random number generator 107 and stored in a T register 108 in the cryptographic facility and also presented as an output so that it can be sent to the
40   smart card. In carrying out the computer/smart card together with the parameters P1 (password) and P2 (program number // diskette serial number, where // denotes concatenation) and a third parameters P3. Where the crypto facility uses the DES algorithm, as shown in Figures 8 and 9, to encrypt the file key, P3 is the computer number. Alternatively, where the crypto facility uses the PK algorithm to encrypt the file key, as shown in Figures 13 and 14, P3 represents the concatenation of the public key of the computer, PKt, and
45   a non-secret constant of sufficient bits which may have a value of zero all decrypted under the secret key, SKu, of the distribution centre.

Also with the smart card, a third operation shown in Figure 17 accepts a password P4 from the smart card provided that the latch 106 is set on. Otherwise, the request is ignored. From the password P4 and the stored random number T in register 108, it derives a file key, KF, that is used by the cryptographic facility
50   to decrypt an encrypted program for the purpose of only executing that program. The decryption algorithm for deriving the file key, KF, may be DES or may be PK. The user has no access to the file key. After the latch 106 has been tested, latch 106 is reset by operation control unit 102. This ensures that a new random number T must be generated before another password will be accepted by the cryptographic facility via another invocation of the third operation and thus prevents old passwords from being played back into the
55   cryptographic facility. Used together, the second and third operations are such that they allow an encrypted program to be decrypted and executed at any computer with a similarly installed cryptographic facility supporting those operations. The microcode for the third operation proceeds as follows: First, latch 106 is tested to see if it is set. If not, the operation is aborted; otherwise, the latch is reset. The parameter P4 is

then decrypted with the "burned in" key KT in decryption block 105 to produce the cipher text output C1. This decryption step is performed with the DES algorithm using KT = eKT(TR5678) where only DES is available as shown in Figure 8 or, with the PK algorithm, the decryption step is performed using KT = SKt where both DES and PK are available as shown in Figure 13. In either case, the resulting cipher text C1 is
5 exclusive ORed with the random number T stored in register 108 to produce the cipher text C2 representing the file key, KF.

The register 104 in addition to storing the address for the decoded microcode, has three flags denoted D, E and P. Up to this point in the description, the operation of the cryptographic facility has been to derive the file key that can be used to decrypt an encrypted program. As will be understood from the following
10 description, it is also possible for a user to use the file key to encrypt data. The flags D, E and P are used to control these operations. The D flag is the decryption flag, and the E flag is the encryption flag. The cryptographic operations of encipher and decipher data are assumed to be such that a cipher operation will be performed only if the E flag is tested and found to be set for an encipher data operation or if the D flag is tested and found to be set for a decipher operation. In addition, before decrypted data is directed from
15 the cryptographic facility, the cryptographic facility will test the P flag. If the P flag is set, the decrypted data will be directed to an execute only memory in the cryptographic facility; otherwise, the decrypted data will be directed to the main memory. Thus, in the case of the first and third operations described above, the microcodes for those operations would in addition set the D flag, reset the E flag, and set the P flag.

In addition to the first three operations, the cryptographic facility also supports a limited set of general
20 purpose cryptographic operations. The first of these, referred to as the fourth operation, is similar to the first operation and is illustrated in Figure 18. With no smart card, the fourth operation accepts user selected parameters P1 and P2, which may be any different arbitrary values selected by the user. From these parameters, the fourth operation generates a file key, KF1, using a different algorithmic procedure than that used by the first operation, where KF1 is a variant of the file key KF. In the fourth operation, the "burned in"
25 key eKT(TR5678) is used to encrypt parameter P2 in encryption block 103. The output of the encryption block 103 is the cipher text C1 which is used in decryption block 105 to produce the cipher text C2 representing KF. KF is then exclusive ORed with the non-zero constant, C, to produce the variant file key KF1. The key KF1 is used only by the cryptographic facility to encrypt and decrypt data. The encrypted and decrypted data are under the control of and accessible to the computer user. In this case, the user has
30 a limited encrypt and decrypt feature, except that the process is performed under the control of a key unknown to the user of the computer. By remembering the parameters P1 and P2, the computer user can decrypt encrypted data at any later time on his or her computer, or any other computer with a similarly installed cryptographic facility supporting the second and fifth operations, by issuing the second operation to generate a new random number T. The computer/smart card protocol passes T, P1, P2, and P3 to the
35 smart card to generate P4 as shown in Figure 9 or Figure 14.

Then a fifth operation is called for to cause the cryptographic facility to recover the key KF1 from P4, C and the stored random number T. Thus, the user cannot migrate the encrypted data to another computer and decrypt it with the same parameters P1 and P2.

With the smart card, a fifth operation shown in Figure 19 accepts a password P4 from the smart card
40 provided that the latch 106 is set on. Otherwise, the request is ignored. From this parameter P4, the fifth operation generates the file key, KF1, using a different algorithmic procedure than used by the third operation. The password P4 used in the fifth operation is produced by the smart card, using the DES algorithm procedure shown in Figure 9 or the PK algorithm procedure shown in Figure 14, from parameters P1, P2 and P3 as well as the random number T generated by the second operation, where P1 and P2 are
45 user defined parameters and P3 is the computer number when the crypto facility uses the DES algorithm to encrypt KF or P3 is the cryptographic variable eSKu(PKt,0) when the crypto facility uses the PK to encrypt KF. Thus, again the user has a limited encrypt and decrypt feature, except that the process is performed under the control of a key unknown to the user of the computer. By remembering the parameters P1 and P2, the computer user can decrypt encrypted data at any later time on his or her computer, or any other
50 computer with a similarly installed cryptographic facility supporting the second and fifth operations, by issuing the second operation to generate a new random number T. The computer/smart card protocol passes T, P1, P2 and P3 to the smart card to generate P4 as shown in Figure 9 or Figure 14. Then a fifth operation is called for to cause the cryptographic facility to recover the key KF1 from P4, C and the stored random number T. Thus, the user can migrate the encrypted data to another computer and decrypt it with
55 the same parameters P1 and P2. Again, the encrypted and decrypted data are under the control of and accessible to the computer user. The microcode for the fifth operation proceeds as follows: First, latch 106 is tested to see if it is set on. If it is not, the operation is aborted; otherwise, the latch is reset and the input parameter P4 is decrypted in decryption block 105 with the "burned in" key KT. Where only DES is

10

available, KT = eKT(TR5678), but where both DES and PK are available., KT = SKt. The output of the decryption block 105 is the cipher text C1 which is exclusive ORed with the random number T stored in register 108 to produce the cipher text C2 representing KF. KF is then exclusive ORed with the non-zero constant C to produce the variant file key KF1. The microcode then sets the D and E flags and resets the P flag in the register 104.

With no smart card, a sixth operation shown in Figure 20 accepts a user selected password P1 and a number P2, which may be any arbitrary values selected by the user, and from these numbers, it generates a file key, KF, using the same algorithmic procedure as used by the first operation. The file key is used only by the cryptographic facility to encrypt data. In this case, the user can encrypt his or her own data but not decrypt it. Used in conjunction with the first operation, the sixth operation allows a user to encrypt his or her own programs and store them on diskette or hard disk in protected form. The parameters P1 and P2 can also be written in the header record of the diskette file or disk file. Later, the saved values of P1 and P2 are used as input parameters with the first operation to decrypt and execute programs. P1 and P2 are such that they permit the encrypted program to be decrypted and executed only at the computer where the program was originally encrypted, so that encrypted programs cannot be migrated to other computers and executed. The microcode for the sixth operation proceeds as follows: First, the parameter P2 is encrypted in encryption block 103 with the "burned in" key eKT(TR5678) to produce the cipher text C1. The cipher text C1 is used in decryption block 105 to decrypt the parameter P1 and produce the cipher text C2 representing the file key, KF. The microcode sets the E flag and resets the D and P flags in register 104.

With a smart card, a seventh operation shown in Figure 21 accepts a password P4 from the smart card provided that the latch 106 is set on. Otherwise, the request is ignored. From this it generates a file key, KF, using the same algorithmic procedure used by the third operation. More specifically, the microcode decrypts the input parameter P4 in decryption block 105 with the "burned in" key KT to produce the cipher text C1. With the DES algorithm alone, KT = eKT(TR5678) is used, or with a PK algorithm, KT = SKt. The cipher text is then exclusive ORed with the random number T stored in register 108 to produce the cipher text C2 representing the file key, KF. The microcode for the seventh operation also sets the E flag and resets the D and P flags in the register 104. In the seventh operation, the key KF is used only by the cryptographic facility to encrypt data. The user can encrypt his or her own data but cannot decrypt it. The password P4 used in the seventh operation is produced by the smart card, using the DES algorithm procedure shown in Figure 9 or the PK algorithm procedure shown in Figure 14, from the parameters P1, P2 and P3 as well as the random number T generated by the second operation. P1 and P2 are user defined parameters. Where the crypto facility uses the DES algorithm, as shown in Figures 8 and 9, to encrypt the file key, P3 is the computer number. Alternatively, where the crypto facility uses the PK algorithm to encrypt the file key, as shown in Figures 13 and 14, P3 represents the concatenation of the public key of the computer, PKt, and a nonsecret constant of sufficient bits which may have a value of zero all decrypted under the secret key, SKu, of the distribution centre. Used in conjunction with the second operation, the seventh operation allows the user to encrypt his or her own programs and store them on a diskette or hard disk in protected form. The parameters P1 and P2 can also be written in the header record of the diskette or disk file. Later, the computer user can decrypt and execute the program on his or her computer, or any other computer with a similarly installed cryptographic facility supporting the second and third operations, by issuing the second operation to produce a new random number T, passing parameters P1, P2, P3, and T to the smart card and requesting a new value of P4, and issuing the third operation to recover the file key KF in the cryptographic facility from the parameter P4 and the stored random number T. Thus, the parameters P1 and P2 are such that they permit an encrypted program to be decrypted and executed at other computers supporting the second and third operations. Because the seventh operation does not allow decryption under the recovered key KF, it cannot be misused by a user to decrypt an encrypted program purchased in the usual manner.

Summarizing, the procedures that are available to a user of a computer with a cryptographic facility that supports the seven operations just described are listed in the table below:

DES Only, No Smart Card   Operations

1. Program Decryption-Execution   OP1
2. File Encryption/Decryption   OP4
3. Program Encryption   OP6

DES Only, With Smart Card

1. Program Decryption-Execution   OP2, OP3
2. File Encryption/Decryption   OP2, OP5
   3. Program Encryption   OP2, OP7

DES/PK, With Smart Card

1. Program Decryption-Execution   OP2, OP3
2. File Encryption/Decryption   OP2, OP5
3. Program Encryption   OP2, OP7

Claims

1. A method of software protection comprising the steps of:

   encrypting a program with a file key, and writing the encrypted program on a storage medium as a program file;

   providing a computer, having a protected memory and a cryptographic facility, with a secret cryptographic key identifier cryptographically derived from a number identifying the computer;

   providing a user of the storage medium containing the encrypted program with a password cryptographically derived from at least the file key, the number identifying the computer, a number identifying the storage medium, and a number identifying the program;

   entering the password into the computer when the medium is loaded thereinto;
   deriving the file key from at least the password, the storage medium number, the program number, and the cryptographic key identifier; and

   decrypting into executable form at least a portion of the program using the derived file key.

2. A method as claimed in claim 1 , wherein, only when the program is first loaded in said computer, performing the steps of prompting the user for the password, and in response to the input of the password by the user, writing the password in the header record of the program file.

3. A method as claimed in claim 1 or claim 2, further comprising the step of writing the decrypted program into said protected memory from which it can only be executed.

4. A method as claimed in any preceding claim wherein the step of providing the password is performed by providing the user of the storage medium containing the encrypted program with an authorisation

number, requesting the user to input the authorisation number, the program number, the storage medium number and the number identifying the computer, computing from the inputted program number and storage medium number a computed authorisation number, comparing the computed authorisation number with the inputted authorisation number, providing a key distribution centre with the inputted program number, storage medium number and the number identifying the computer if the computed and inputted authorisation numbers are the same, otherwise rejecting a password request by the user, generating a first key as a function of the number identifying the computer and then encrypting the program number and storage medium number concatenated together with said first key to produce a second key at the key distribution centre, and encrypting the secret file key of the program with said second key to produce said password.

5. A method as claimed in claim 4 further comprising the step of determining if the computed authorisation number has been used before, and if it has not, then performing the step of comparing the computed authorization number with the inputted authorisation number, otherwise rejecting a password request by the user.

6. A method as claimed in claim 2 and any one of claim 4 or claim 5 wherein the steps of deriving the file key and decrypting at least a portion of the program are performed by the steps of reading said password, program number and storage medium number from said header record, encrypting said program number and storage medium number from said header record, encrypting said program number and storage medium number concatenated together with a key which is a function of said secret cryptographic key identifier to produce a decryption key, decrypting said password with said decryption key to produce said file key, and decrypting the program using said secret file key.

7. A method as claimed in claim 1 or claim 2, wherein the step of providing a computer with a secret cryptographic key identifier is performed by the software or computer vendor issuing to the user a smart card having said secret cryptographic key identifier, said smart card interfacing with said computer.

8. A method as claimed in claim 7 wherein the step of providing the password is performed by the steps of providing the user of the storage medium containing the encrypted program with an authorization number, requesting the user to input the authorisation number, a number of the smart card, the program number and the storage medium number, computing an authorisation number from the inputted program number and storage medium number, comparing the computed authorisation number with the inputted authorization number, and if the computed and inputted authorisation numbers are the same, providing a key distribution centre with the inputted number of the smart card, the program number and the storage medium number, otherwise rejecting a password request by the user, generating a card key corresponding to the inputted card number and then encrypting the program number and storage medium number concatenated together with said card key to produce an encryption key at the key distribution centre, generating a secret file key corresponding to the inputted program number, and encrypting said secret file key with said encryption key to produce said password.

9. A method as claimed in claim 7 or claim 8, wherein the steps of deriving the file key and decrypting at least a portion of the program, are performed by the steps of supplying the smart card with the password, program number and storage medium number, encrypting in the smart card the program number and storage medium number concatenated together with the key of the smart card to produce a decryption key, decrypting the password with said decryption key to produce the secret file key, and decrypting in the computer at least said portion of the program using the secret file key.

10. A method as claimed in claim 9 further comprising the steps of supplying the smart card with a number identifying said cryptographic facility, encrypting in the smart card the number identifying said cryptographic facility with a universal key to produce a computer encryption key, generating in the computer a random number and supplying the random number to the smart card, exclusive ORing in the smart card the secret file key with the random number and encrypting the result with said computer encryption key to produce an encrypted exclusive ORed output, decrypting in the computer the encrypted exclusive ORed output with the computer encryption key, and exclusive ORing the decrypted exclusive ORed output with said random number to produce the secret file key.

13

11. A method as claimed in claim 10 wherein the steps of encryption in the smart card and decryption in the computer are performed using the DES algorithm.

12. A method as claimed in claim 9 further comprising the steps providing the computer with a public key, PKt, decrypted under the secret key of a public registry, and also providing said cryptographic facility with a corresponding secret key, SKt, providing said smart card with a public key, PKu, encrypting in the smart card the computer public key decrypted under the secret key of the public registry with the card's public key PKu to produce said key PKt, generating in the computer a random number and supplying the random number to the smart card, exclusive ORing in the smart card the secret file key with the random number and encrypting the result with said key PKt to produce an encrypted exclusive ORed output, decrypting in the computer the encrypted exclusive ORed output with the key SKt, and exclusive ORing the decrypted exclusive ORed output with said random number to produce the secret file key.

13. A method as claimed in claim 12 wherein the steps of encryption and decryption in the smart card and the computer are performed by selectively using the DES algorithm and a public key algorithm.

14. A method as claimed in claim 1 wherein said cryptographic facility supports encryption and decryption of user generated data comprising the steps of accepting a user input parameter, and decrypting said user input parameter under a key which is a function of said secret unique cryptographic key identifier to generate a file key for encrypting and decrypting said user generated data.

15. A method as claimed in claim 14 wherein said input parameter is input as a first parameter and a second parameter, said step of decrypting comprises the steps of encrypting said second parameter under said key which which is a function of said secret unique cryptographic key identifier to produce a first cipher text, and decrypting said first parameter using said first cipher text to produce a second cipher text corresponding to a key KF.

16. A method as claimed in claim 14 further comprising the step of exclusive ORing said key KF with a constant to produce a key KF1 which is used as a file key for encrypting and decrypting said user generated data.

17. A method as claimed in claim 14 further comprising the steps of generating a random number, and exclusive ORing said random number with the decrypted user input parameter to produce a key KF and exclusive ORing said key KF with a constant to produce a key KF1 which is used for encrypting and decrypting user generated data.

18. A method as claimed in claim 14 wherein said cryptographic facility supports encryption of user generated data comprising the steps of encrypting user generated data under said file key, decrypting user generated data by the steps of accepting the user input parameter in the form of a first parameter and a second parameter, encrypting said second parameter with said key which is a function of said secret cryptographic key identifier to produce a decrytion key, decrypting said first parameter with said decryption key to produce a key KF related to said file key, and decrypting said user generated data using said file key.

Patentansprüche

1. Verfahren zur Sicherung von Software, das die folgenden Schritte aufweist:

Verschlüsseln eines Programmes mit einem Dateischlüssel und Schreiben des verschlüsselten Programmes als Programmdatei auf ein Speichermedium;

Bereitstellen eines Computers, der einen geschützten Speicher und eine kryptographische Einrichtung besitzt mit einer Erkennung für eine geheime kryptographischen Schlüsselerkennung, der kryptographisch aus einer den Computer identifizierenden Nummer hergeleitet wird;

Ausstatten eines Benutzers des Speichermediums, welches das verschlüsselte Programm enthält, mit einem Kennwort, das kryptographisch zumindest aus dem Datenschlüssel, der den Computer identifi-

zierenden Nummer, einer das Speichermedium identifizierenden Nummer und einer das Programm identifizierenden Nummer hergeleitet wird;

Eingehen des Kennwortes in den Computer, wenn das Medium darin geladen ist;

Herleiten des Dateischlüssels von zumindest dem Kennwort, der Nummer des Speichermediums, der Programmnummer und der kryptographischen Schlüsselerkennung und

Entschlüsseln zumindest eines Teilabschnittes des Programmes in eine ausführbare Form unter Verwendung des Dateischlüssels.

2. Verfahren nach Anspruch 1, bei welchem die Schritte der Eingabeaufforderung des Kennwortes an den Benutzer nur ausgeführt werden, wenn zuerst das Programm in den Computer geladen wurde, und bei welchem als Reaktion auf die Eingabe des Kennwortes durch den Benutzer das Kennwort in den Kopfdatensatz der Programmdatei geschrieben wird.

3. Verfahren nach Anspruch 1 oder 2, das weiters den Schritt zum Schreiben des entschlüsselten Programmes in den geschützten Speicher aufweist, aus dem es ausschließlich ausgeführt werden kann.

4. Verfahren nach einem der vorigen Ansprüche, bei welchem der Schritt zum Bereitstellen des Kennwortes durch Ausstatten des Benutzers des Speichermediums, welches das verschlüsselte Programm enthält, mit einer Autorisierungsnummer, Auffordern des Benutzers zur Eingabe der Autorisierungsnummer, der Programmnummer, der Nummer des Speichermediums und der den Computer identifizierenden Nummer, Berechnen einer berechneten Autorisierungsnummer aus der eingegebenen Programmnummer und der Nummer des Speichermediums, Vergleichen der berechneten Autorisierungsnummer mit der eingegebenen Autorisierungsnummer, Beliefern eines Schlüsselverteilungszentrums mit der eingegebenen Programmnummer, Nummer des Speichermediums und der den Computer identifizierenden Nummer, falls die berechnete und die eingegebene Autorisierungsnummer gleich sind, andernfalls Zurückweisen einer Kennwortanforderung des Benutzers, Erzeugen eines ersten Schlüssels als Funktion der den Computer identifizierenden Nummer und dann Verschlüsseln der Programmnummer und der Nummer des Speichermediums, die mit dem ersten Schlüssel miteinander verkettet sind, um in dem Schlüsselverteilungszentrum einen zweiten Schlüssel zu erzeugen, und Verschlüsseln des geheimen Dateischlüssels des Programmes mit dem zweiten Schlüssel, um das Kennwort zu erzeugen, ausgeführt wird.

5. Verfahren nach Anspruch 4, das weiters den Schritt des Bestimmens, ob die berechnete Autorisierungsnummer zuvor Benutzt wurde, und wenn das nicht der Fall ist, dann das Ausführen des Schrittes zum Vergleichen der berechneten Autorisierungsnummer mit der eingegebenen Autorisierungsnummer, und andernfalls das Zurückweisen der Kennwortanforderung des Benutzers aufweist.

6. Verfahren nach Anspruch 2 und einem der Ansprüche 4 oder 5, bei welchem die Schritte zum Herleiten des Dateischlüssels und zum Entschlüsseln zumindest eines Teilabschnittes des Programmes durch die folgenden Schritte ausgeführt werden: Lesen des Kennwortes, der Programmnummer und der Nummer des Speichermediums vom Kopfdatensatz, Verschlüsseln der Programmnummer und der Nummer des Speichermediums vom Kopfdatensatz, Verschlüsseln der Programmnummer und der Nummer des Speichermediums, die miteinander mit einem Schlüssel verkettet sind, der eine Funktion der geheimen kryptographischen Schlüsselerkennung ist, um einen Entschlüsselungsschlüssel zu erzeugen, Entschlüsseln des Kennwortes mit dem Entschlüsselungsschlüssel, um den Dateischlüssel zu erzeugen, und Entschlüsseln des Programmes unter Verwendung des geheimen Dateischlüssels.

7. Verfahren nach Anspruch 1 oder 2, bei welchem der Schritt des Bereitstellens eines Computers mit einer geheimen kryptographischen Schlüsselerkennung von dem Software- oder dem Computerhändler durch Ausgeben einer Smart-Karte durchgeführt wird, welche die geheime kryptographische Schlüsselerkennung besitzt, wobei die Smart-Karte an den Computer angeschlossen wird.

8. Verfahren nach Anspruch 7, bei welchem der Schritt zum Bereitstellen des Kennwortes durch die folgenden Schritte ausgeführt wird: Ausstatten des Benutzers des Speichermediums, welches das verschlüsselte Programm enthält, mit einer Autorisierungsnummer, Auffordern des Benutzers zur

Eingabe der Autorisierungsnummer, einer Nummer der Smart-Karte, der Programmnummer und der Nummer des Speichermediums, Berechnen einer Autorisierungsnummer aus der eingegebenen Programmnummer und der Nummer des Speichermediums, Vergleichen der berechneten Autorisierungsnummer mit der eingegebenen Autorisierungsnummer, und falls die berechnete und die eingegebene Autorisierungsnummer gleich sind, Beliefern eines Schlüsselverteilungszentrums mit der eingegebenen Nummer der Smart-Karte, der Programmnummer und der Nummer des Speichermediums, andernfalls Zurückweisen einer Kennwortanforderung des Benutzers, Erzeugen eines Kartenschlüssels entsprechend der eingegebenen Kartennummer und dann Verschlüsseln der Programmnummer und der Nummer des Speichermediums, die miteinander durch den Kartenschlüssel verkettet sind, um im Schlüsselverteilungszentrum einen Verschlüsselungsschlüssel zu erzeugen Erzeugen eines geheimen Dateischlüssels entsprechend der eingegebenen Programmnummer und Verschlüsseln des geheimen Dateischlüssels mit dem Verschlüsselungsschlüssel, um das Kennwort zu erzeugen.

9. Verfahren nach Anspruch 7 oder 8, bei welchem die Schritte zum Herleiten des Dateischlüssels und zum Entschlüsseln zumindest eines Teilabschnittes des Programmes durch die folgenden Schritte ausgeführt wird: Beliefern der Smart-Karte mit dem Kennwort, der Programmnummer und der Nummer des Speichermediums, Verschlüsseln der Programmnummer und der Nummer des Speichermediums in der Smart-Karte, die miteinander durch den Schlüssel der Smart-Karte verkettet sind, um einen Entschlüsselungsschlüssel zu erzeugen, Entschlüsseln des Kennwortes mit dem Entschlüsselungsschlüssel, um einen geheimen Dateischlüssel zu erzeugen, und Entschlüsseln zumindest den Teilabschnitt des Programmes in dem Computer unter Verwendung des geheimen Dateischlüssels.

10. Verfahren nach Anspruch 9, das weiters die folgenden Schritte aufweist: Beliefern der Smart-Karte mit einer die kryptographische Einrichtung identifizierenden Nummer, Verschlüsseln der die kryptographische Einrichtung identifizierenden Nummer mit einem universellen Schlüssel in der Smart-Karte, um einen Computerverschlüsselungsschlüssel zu erzeugen, Erzeugen einer Zufallszahl im Computer und Beliefern der Smart-Karte mit der Zufallszahl, Exklusiv-ODER-Verknüpfen des geheimen Dateischlüssels mit der Zufallszahl in der Smart-Karte und Verschlüsseln des Ergebnisses mit dem Computerverschlüsselungsschlüssel, um eine verschlüsselte Exklusiv-ODER-Ausgabe zu erzeugen, Entschlüsseln der verschlüsselten Exklusiv-ODER-Ausgabe mit einem Computerverschlüsselungsschlüssel in dem Computer und Exklusiv-ODER-Verknüpfen der entschlüsselten Exklusiv-ODER-Ausgabe mit der Zufallszahl, um den geheimen Dateischlüssel zu erzeugen.

11. Verfahren nach Anspruch 10, bei welchem die Schritte des Verschlüsselns in der Smart-Karte und des Entschlüsselns im Computer unter Verwendung des DES-Algorithmus durchgeführt werden.

12. Verfahren nach Anspruch 9, das weiters die folgenden Schritte aufweist: Beliefern des Computers mit einem öffentlichen Schlüssel, PKt, der mit dem geheimen Schlüssel einer öffentlichen Registrierung entschlüsselt wurde, und ebenso Beliefern der kryptographischen Einrichtung mit einem entsprechenden geheimen Schlüssel, SKt, Beliefern der Smart-Karte mit einem öffentlichen Schlüssel, PKu, Verschlüsseln des öffentlichen Schlüssels des Computers in der Smart-Karte, der mit dem geheimen Schlüssel der öffentlichen Registrierung entschlüsselt wurde, mit dem öffentlichen Schlüssel PKu der Karte, um den Schlüssel PKt zu erzeugen, Erzeugen einer Zufallszahl im Computer und Liefern der Zufallszahl an die Smart-Karte, Exclusiv-ODER-Verknüpfen des geheimen Dateischlüssels mit der Zufallszahl in der Smart-Karte und Verschlüsseln des Ergebnisses mit dem Schlüssel PKt, um eine verschlüsselte Exklusiv-ODER-Ausgabe zu erzeugen, Entschlüsseln der verschlüsselten Exklusiv-ODER-Ausgabe im Computer mit dem Schlüssel SKt und Exklusiv-ODER-Verknüpfen der entschlüsselten Exklusiv-ODER Ausgabe mit der Zufallszahl, um den geheimen Dateischlüssel zu erzeugen.

13. Verfahren nach Anspruch 12, bei welchem die Schritte des Verschlüsselns und des Entschlüsselns in der Smart-Karte und im Computer durch selektives Verwenden des DES-Algorithmus und eines öffentlichen Schlüsselalgorithmus ausgeführt werden.

14. Verfahren nach Anspruch 1, bei welchem die kryptographische Einrichtung das Verschlüsseln und Entschlüsseln von Daten unterstützt, die vom Benutzer erzeugt werden, das die Schritte des Annehmens eines Benutzereingabeparameters und des Entschlüsselns des Benutzereingabeparameters mit einem Schlüssel aufweist, der eine Funktion der geheimen eindeutigen kryptographischen Schlüsselerkennung ist, um einen Dateischlüssel zum Verschlüsseln und Entschlüsseln der vom Benutzer erzeug-

ten Daten zu erzeugen.

15. Verfahren nach Anspruch 14, bei welchem der Eingabeparameter als erster und als zweiter Parameter eingegeben wird, wobei der Schritt des Entschlüsselns die folgenden Schritte aufweist: Verschlüsseln des zweiten Parameters mit dem Schlüssel, der eine Funktion der geheimen eindeutigen kryptographischen Schlüsselerkennung ist, um einen ersten Chiffretext zu erzeugen, und Entschlüsseln des ersten Parameters unter Verwendung des ersten Chiffretextes, um einen zweiten Chiffretext zu erzeugen, der einem Schlüssel KF entspricht.

16. Verfahren nach Anspruch 14, das weiters den Schritt des Exklusiv-ODER-Verknüpfens des Schlüssels KF mit einer Konstante aufweist, um einen Schlüssel KF1 zu erzeugen, der als Dateischlüssel zum Verschlüsseln und Entschlüsseln der vom Benutzer erzeugten Daten verwendet wird.

17. Verfahren nach Anspruch 14, das weiters die Schritte des Erzeugens einer Zufallszahl und des Exklusiv-ODER-Verknüpfens der Zufallszahl mit dem entschlüsselten Benutzereingabeparameter, um einen Schlüssel KF zu erzeugen, und des Exklusiv-ODER-Verknüpfens des Schlüssels KF mit einer Konstante aufweist, um einen Schlüssel KF1 zu erzeugen, der zum Verschlüsseln und Entschlüssein der vom Benutzer erzeugten Daten verwendet wird.

18. Verfahren nach Anspruch 14, bei welchem die kryptographische Einrichtung das Verschlüsseln von Daten unterstützt, die vom Benutzer erzeugt werden, das die folgenden Schritte aufweist: Verschlüsseln vom Benutzer erzeugter Daten mit dem Dateischlüssel, Entschlüsseln der vom Benutzer erzeugten Daten durch die Schritte des Annehmens des Benutzereingabeparameters in Form eines ersten und eines zweiten Parameters, Verschlüsseln des zweiten Parameters mit dem Schlüssel, der eine Funktion der geheimen kryptographischen Schlüsselerkennung ist, um einen Entschlüsselungsschlüssel zu erzeugen, Entschlüsseln des ersten Parameters mit dem Entschlüsselungsschlüssel, um einen Schlüssel KF zu erzeugen, der mit dem Dateischlüssel in Verbindung steht, und Entschlüsseln der vom Benutzer erzeugten Daten unter Verwendung des Dateischlüssels.

Revendications

1. Procédé de protection de logiciel, comprenant les étapes de:
chiffrer un programme avec une clé de fichier, et écrire le programme chiffré sur un support d'emmagasinage en tant qu'un fichier de programme ;
pourvoir un ordinateur, ayant une mémoire protégée et une facilité cryptographique, d'un identificateur de clé cryptographique secrète dérivé cryptographiquement d'un numéro identifiant l'ordinateur;
pourvoir un utilisateur du support d'emmagasinage contenant le programme chiffré, d'un mot de passe cryptographiquement dérivé au moins de la clé de fichier, du numéro identifiant l'ordinateur, d'un numéro identifiant le support d'emmagasinage, et d'un numéro identifiant le programme;
entrer le mot de passe dans l'ordinateur lorsque le support y est chargé;
dériver la clé de fichier au moins du mot de passe, du numéro de support d'emmagasinage, du numéro de programme, et de l'identificateur de clé cryptographique; et
déchiffrer en forme exécutable au moins une portion du programme utilisant la clé de fichier dérivée.

2. Procédé selon la revendication 1 dans lequel, seulement lorsque le programme est d'abord chargé dans ledit ordinateur, il est accompli les étapes de sollicitation du mot de passe de l'utilisateur et, en réponse à l'entrée du mot de passe par l'utilisateur, d'écriture du mot de passe dans l'enregistrement d'en-tête du fichier de programme

3. Procédé selon les revendications 1 ou 2, comprenant en outre l'étape d'écriture du programme déchiffré dans ladite mémoire protégée à partir de laquelle il peut être seulement exécuté.

4. Procédé selon l'une quelconque des revendications précédentes, dans lequel l'étape de pourvoir le mot de passe est accomplie en fournissant à l'utilisateur du support d'emmagasinage contenant le programme chiffré un numéro d'autorisation demandant à l'utilisateur d'entrer le numéro autorisé, le numéro de programme, le numéro de support d'emmagasinage et le numéro identifiant l'ordinateur, en calculant à partir du numéro de programme et du numéro de support d'emmagasinage entrés, un
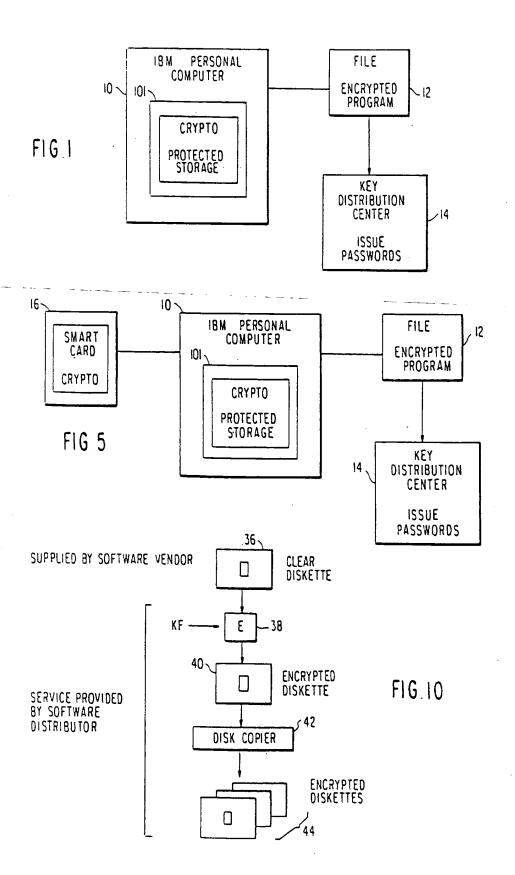
numéro d'autorisation d'ordinateur, en comparant le numéro d'autorisation calculé avec le numéro d'autorisation entré, en fournissant au centre de distribution de clés les numéro de programme, numéro de support d'emmagasinage et numéro identifiant le calculateur entrés si les numéros d'autorisation entré et calculé sont les mêmes, autrement en rejetant une demande de mot de passe par l'utilisateur,
5   engendrant une première clé en fonction du numéro identifiant l'ordinateur et ensuite en chiffrant le numéro de programme et le numéro de support d'emmagasinage enchaînés avec ladite première clé pour produire une deuxième clé au centre de distribution de clés, et en chiffrant la clé de fichier secrète du programme avec ladite deuxième clé pour produire ledit mot de passe.
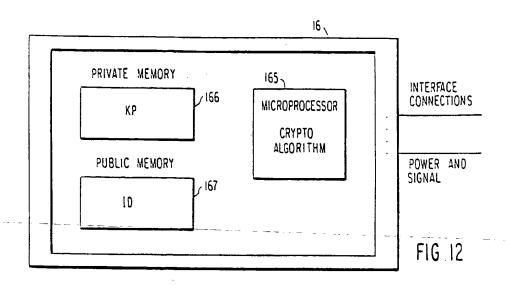
10 **5.**   Procédé selon la revendication 4, comprenant en outre l'étape de déterminer si le numéro d'autorisation calculé a été utilisé auparavant et, dans la négative, accomplir alors l'étape de comparer le numéro d'autorisation calculé avec le numéro d'autorisation entré, autrement, rejeter une demande de mot de passe par l'utilisateur.

15 **6.**   Procédé selon la revendication 2 et selon l'une quelconque des revendications 4 ou 5, dans lequel les étapes de dériver la clé de fichier et déchiffrer au moins une portion du programme, sont accomplies par les étapes d'extraire lesdits mot de passe, numéro de programme et numéro de support d'emmagasinage dudit enregistrement d'en-tête, chiffrer lesdits numéro de programme et numéro de support d'emmagasinage à partir de l'enregistrement d'en-tête, chiffrer lesdits numéro de programme
20   et numéro de support d'emmagasinage enchaînés avec une clé qui est fonction dudit identificateur de clé cryptographique secrète pour produire la clé de déchiffrement, déchiffrer ledit mot de passe avec ladite clé de déchiffrement pour produire ladite clé de fichier, et déchiffrer le programme utilisant ladite clé de fichier secrète.

25 **7.**   Procédé selon les revendications 1 ou 2, dans lequel l'étape de pourvoir un ordinateur d'un identificateur de clé cryptographique secrète est accomplie par le vendeur d'ordinateur ou de logiciel délivrant à l'utilisateur une carte à mémoire ayant ledit identificateur de clé cryptographique secrète, ladite carte à mémoire faisant interface avec ledit ordinateur.

30 **8.**   Procédé selon la revendication 7, dans lequel l'étape de pourvoir un mot de passe est accomplie par les étapes de fournir à l'utilisateur du support d'emmagasinage contenant le programme chiffré, un numéro d'autorisation, demander à l'utilisateur d'entrer le numéro d'autorisation, numéro de la carte à mémoire, le numéro de programme et le numéro de support d'emmagasinage, calculer un numéro d'autorisation à partir du numéro de programme entré et du numéro de support d'emmagasinage,
35   comparer le numéro d'autorisation calculé avec le numéro d'autorisation entré, et si les numéros d'autorisation calculé et entré sont les mêmes, fournir au centre de distribution de clés le numéro entré de la carte à mémoire, le numéro de programme et le numéro de support d'emmagasinage, autrement, rejeter une demande de mot de passe par l'utilisateur, engendrer une clé de carte correspondant au numéro de carte entré et ensuite chiffrer le numéro de programme et le numéro de support
40   d'emmagasinage enchaînés avec ladite clé de carte pour produire une clé de chiffrement au centre de distribution de clés, engendrer une clé de fichier secrète correspondant au numéro de programme entré, et chiffrer ladite clé de fichier secrète avec ladite clé de chiffrement pour produire ledit mot de passe.

45 **9.**   Procédé selon les revendications 7 ou 8, dans lequel les étapes de dériver la clé de fichier et déchiffrer au moins une portion du programme sont accomplies par les étapes de fournir à la carte à mémoire le mot de passe, le numéro de programme et le numéro de support d'emmagasinage, chiffrer dans la carte à mémoire le numéro de programme et le numéro de support d'emmagasinage enchaînés avec la clé de la carte à mémoire pour produire une clé de déchiffrement, déchiffrer le mot de passe avec
50   ladite clé de déchiffrement pour produire la clé de fichier secrète, et déchiffrer dans l'ordinateur au moins ladite portion du programme utilisant la clé de fichier secrète.

 10. Procédé selon la revendication 9, comprenant en outre les étapes de fournir à la carte à mémoire un numéro identifiant ladite facilité cryptographique, chiffrer dans la carte à mémoire le numéro identifiant
55   ladite facilité cryptographique avec une clé universelle pour produire une clé de chiffrement d'ordinateur, engendrer dans l'ordinateur un numéro aléatoire et fournir le numéro aléatoire à la carte à mémoire, soumettre à une condition OU exclusif dans la carte à mémoire la clé de fichier secrète avec le numéro aléatoire, et chiffrer le résultat avec ladite clé de chiffrement d'ordinateur pour produire une

18

sortie soumise à une condition OU exclusif chiffrée, déchiffrer dans l'ordinateur la sortie soumise à une condition OU exclusif chiffrée avec la clé de chiffrement d'ordinateur, et soumettre à une condition OU exclusif la sortie soumise à une condition OU exclusif déchiffrée avec ledit numéro aléatoire pour produire la clé de fichier secrète.

11. Procédé selon la revendication 10, dans lequel les étapes de chiffrement dans la carte à mémoire et de déchiffrement dans l'ordinateur sont accomplies en utilisant l'algorithme DES.

12. Procédé selon la revendication 9, comprenant en outre les étapes de pourvoir l'ordinateur d'une clé publique, PKt, déchiffrée sous la clé secrète d'un registre public, et également de pourvoir ladite facilité cryptographique d'une clé secrète correspondante, SKt; fournissant à ladite carte à mémoire une clé publique, PKu, chiffrer dans la carte à mémoire la clé publique d'ordinateur déchiffrée sous la clé secrète du registre public avec la clé publique de la carte PKu pour produire ladite clé PKt, engendrer dans l'ordinateur un numéro aléatoire et fournir le numéro aléatoire à la carte à mémoire, soumettre à une condition OU exclusif dans la carte à mémoire la clé de fichier secrète avec le numéro aléatoire, et chiffrer le résultat avec ladite clé PKt pour produire une sortie OU exclusif chiffrée, déchiffrer dans l'ordinateur la sortie OU exclusif chiffrée avec la clé SKt, et soumettre à une condition OU exclusif la sortie OU exclusif déchiffrée avec le numéro aléatoire pour produire la clé de fichier secrète.

13. Procédé selon la revendication 12, dans lequel les étapes de chiffrement et de déchiffrement dans la carte à mémoire et dans l'ordinateur sont accomplies en utilisant sélectivement l'algorithme DES et un algorithme de clé publique.

14. Procédé selon la revendication 1, dans lequel ladite facilité cryptographique assiste le chiffrement et le déchiffrement des données engendrées par l'utilisateur comprenant les étapes d'accepter un paramètre d'entrée d'utilisateur, et de déchiffrer ledit paramètre d'entrée d'utilisateur sous une clé qui est fonction dudit identificateur de clé cryptographique secrète unique afin d'engendrer une clé de fichier pour chiffrer et déchiffrer lesdites données engendrées par l'utilisateur.

15. Procédé selon la revendication 14, dans lequel ledit paramètre d'entrée est entré en tant que premier paramètre et deuxième paramètre, ladite étape de déchiffrement comprenant les étapes de chiffrer ledit deuxième paramètre sous ladite clé qui est fonction dudit identificateur de clé cryptographique secrète unique pour produire un premier texte en chiffre, et de déchiffrer ledit premier paramètre en utilisant ledit premier texte en chiffre pour produire un deuxième texte en chiffre correspondant à une clé KF.

16. Procédé selon la revendication 14, comprenant en outre l'étape de soumettre à une condition OU exclusif ladite clé KF avec une constante pour produire une clé KF1 qui soit utilisée en tant que clé de fichier pour chiffrer et déchiffrer lesdites données engendrées par l'utilisateur.

17. Procédé selon la revendication 14, comprenant en outre les étapes d'engendrer un numéro aléatoire et de soumettre à une condition OU exclusif ledit numéro aléatoire avec le paramètre d'entrée d'utilisateur déchiffré pour produire une clé KF1 qui soit utilisée pour chiffrer et déchiffrer les données engendrées par l'utilisateur.

18. Procédé selon la revendication 14, dans lequel ladite facilité cryptographique assiste le chiffrement des données engendrées par l'utilisateur comprenant les étapes de chiffrer les données engendrées par l'utilisateur sous ladite clé de fichier, déchiffrer les données engendrées par l'utilisateur par les étapes d'accepter le paramètre d'entrée de l'utilisateur sous forme d'un premier paramètre et d'un deuxième paramètre, chiffrer ledit deuxième paramètre avec ladite clé qui est fonction dudit identificateur de clé cryptographique secrète pour produire une clé de déchiffrement, déchiffrer ledit premier paramètre avec ladite clé de déchiffrement pour produire une clé KF associée à ladite clé de fichier, et déchiffrer lesdites données engendrées par l'utilisateur en utilisant ladite clé de fichier.

FIG.1

FIG 5

SUPPLIED BY SOFTWARE VENDOR

SERVICE PROVIDED
BY SOFTWARE
DISTRIBUTOR

FIG.10

16

PRIVATE MEMORY

KP /166

PUBLIC MEMORY

ID /167

165

MICROPROCESSOR

CRYPTO
ALGORITHM

INTERFACE
CONNECTIONS

POWER AND
SIGNAL

FIG 12

(DISKETTE)

FIG 2

IBM CORPORATION
PROGRAM # 12
DISKETTE # 3456

COMPUTER PASSWORD (SUPPLIED BY SOFTWARE VENDOR
AND WRITTEN ON FILE (DISKETTE OR HARD DISK)
BY INITIALIZATION PROGRAM (REWRITE/OVERWRITE
NOT PERMITTED BY SOFTWARE)

eKTPG123456(KF)

ENCRYPTED PROGRAM:
F C D 8 3 2 9 1 D E 2 1 F 0 C 0
8 6 9 E F D 7 F 9 F 2 6 5 A 0 9
8 8 D 5 5 E 5 4 F 5 4 C 9 7 B 4
0 C 0 C C 0 0 C 8 3 E A 4 8 F D
8 3 B C 8 E R 3 A 6 5 7 0 1 8 3
D F 7 2 5 D C A D 9 4 E A 2 E 9
E 6 5 2 B 5 3 B 5 5 0 B E 8 B 0
A F 5 2 7 1 2 0 C 4 8 5 C B B 0
0 F 0 4 C E 3 9 3 D B 9 2 6 D 5
C 9 F 0 0 F F C 7 4 0 7 9 0 6 7
7 C F D 8 2 A 5 9 3 2 5 2 B 4 E

USER   SOFTWARE   VENDOR   FIG.3

KEY DISTRIBUTION
CENTER

FIRST USE   22   N
REISSUE
PASSWORD
AUTH #   AUTH # OK
24   N
Y
COMPUTER #   CONTINUE   REJECT   14

GENERATE   KT   E   26
20   AUTH #   SK
DISK #   21

PROG #

PROG # KEY   28

1   KF1   32
E   30
KTTR

m   KFm
KTPG123456

KF   ENCRYPTION CAN
BE USED FOR
eKTPG123456(KF)   E   34   KEY SECRECY
(PASSWORD)

(FROM FILE HEADER)   10   IBM PC: # 5678
eKTPG123456(KF)(PASSWORD)

12 3456 (PROG #, DISK #)   101

KTTR5678

103
E

FIG.4   105
KTPG123456   D

KF

DES ALGORITHM

(CRYPTOGRAPHIC FACILITY)
(COMPUTER)   (SERIAL # 5678)

(DISKETTE)

```
┌─────────────────────────────────────────────────────┐
│ IBM CORPORATION                                      │
│ PROGRAM # 12                                         │
│ DISKETTE # 3456                                      │
├─────────────────────────────────────────────────────┤
│ SMART CARD PASSWORD ( SUPPLIED BY SOFTWARE VENDOR AND│
│        WRITTEN ON FILE ( DISKETTE OR HARD DISK ) BY  │
│        INITIALIZATION PROGRAM ( REWRITE/OVERWRITE NOT│
│        PERMITTED BY SOFTWARE )                       │
│                                                      │
│ eKPPG123456 (KF)                                     │
├─────────────────────────────────────────────────────┤
│ ENCRYPTED PROGRAM:                                   │
│    F C 0 8 3 2 9 1 0 E 2 1 F 0 C 0                   │
│    8 6 9 E F 0 7 F 9 F 2 6 5 A 0 9                   │
│    8 8 0 5 5 E 5 4 F 5 4 C 9 7 B 4                   │
│    0 C 0 C C 0 0 C 8 3 E A 4 8 F 0                   │
│    8 3 B C 8 E R 3 A 6 5 7 0 1 8 3                   │
│    0 F 7 2 5 0 C A 0 9 4 E A 2 E 9                   │
│    E 6 5 2 B 5 3 B 5 5 0 B E 8 B 0                   │
│    A F 5 2 7 1 2 0 C 4 8 5 C B B 0                   │
│    0 F 0 4 C E 3 9 3 0 B 9 2 6 0 5                   │
│    C 9 F 0 0 F F C 7 4 0 7 9 0 6 7                   │
│    7 C F 0 8 2 A 5 9 3 2 5 2 B 4 E                   │
└─────────────────────────────────────────────────────┘
```

FIG. 7

FIG. 6

USER — SOFTWARE VENDOR — KEY DISTRIBUTION CENTER

eKPPG123456 (KF)
(PASSWORD)

(ENCRYPTION CAN BE USED FOR KEY SECRECY)

(FROM FILE HEADER)

PROG# 12   DISK# 3456   PASSWORD eKPPG123456(KF)

10

IBM PC : #5678

5678   (P3)
123456   (P2)
eKPPG123456(KF)   (P1)

101

RN GENERATOR   107

INTERNAL
PROTOCOL
INTERCHANGE

T

eKTTRS678(KF⊕T)

(P4)

KTTR5678   D   109

T

⊕

KF

DES ALGORITHM

FIG. 8

(CRYPTOGRAPHIC FACILITY)
(COMPUTER) (SERIAL # 5678)

16

IBM SMART CARD : # 1234

KP

KT   E   161

(P3)   5678

E   162

(P2)   123456

KPPG123456

(P1)   eKPPG123456 (KF)
       (CARD PASSWORD)

D   163

⊕   T

KTTR5678   E   164

INTERNAL
PROTOCOL
INTERCHANGE

(P4)
eKTTRS678(KF⊕T)
(COMPUTER PASSWORD)

DES ALGORITHM

FIG. 9

PC BUS

ROM ————————————113————————

COMPUTER "BURNED IN" KEY

ROM MEMORY   114

PROGRAMS (8K)
· KEY MANAGEMENT SEQUENCE GENERATORS
· DISK LOADER PROGRAMS

RAM MEMORY   115

READ ONLY EXTERNAL (64 BYTES)
        (PARAMETER OUTPUT BUFFER)

WRITE ONLY EXTERNAL (64 BYTES)
        (PARAMETER INPUT BUFFER)

EXECUTE ONLY EXTERNAL (64K)
        (ALLOWS DECRYPTED PROGRAMS TO BE EXECUTED)

READ/WRITE INTERNAL USE ONLY (4K)
        (INTERMEDIATE STORAGE OF PARAMETERS, DATA,
        AND KEYS)

CRYPTOGRAPHIC FACILITY

FIG. II

(FROM FILE HEADER)

PROG#   ENCIPHERED
123456   FILE KEY
eKPPG123456 (KF)

10

STORED PARAMETER
dSKu (PKt,0)

dSKu (PKt,0)   (P3)

123456   (P2)

eKPPG123456 (KF) (P1)

101

107   RN
GENERATOR

T

T

ePKt (KF⊕T)   (P4)

105

SKt — D

KF⊕T   ⊕ — KF   DECRYPT
FILE WITH
KF USING

PK ALGORITHM | DES ALGORITHM   DES

(CRYPTOGRAPHIC FACILITY)

(TERMINAL)
(SERIAL # 5678)

FIG.13

IBM SMART CARD : # 1234
KP   PKu

(P3)   dSKu(PKt.O)

161

E

(P2)   123456

162

E

KPPG123456

(P1)   eKPPG123456(KF)

CHECK
REDUNDANCY BITS
(O BITS)

D   163

KF

T

164

E

PK1

(P4)   ePK1(KF⊕T)

PK ALGORITHM	DES ALGORITHM

FIG. 14

FIG.15

(PASSWORD) P1

(PROG#) (DISK#) P2

OP1

101

102

OPERATION
CONTROL
UNIT

KTTR5678

E 103

104

REGISTER D E P

C1

D 105

C2 = KF

DES ALGORITHM

(CRYPTOGRAPHIC FACILITY)

(COMPUTER)

FIG.16

T

OP2

101

TREG 108

107

RANDOM NUMBER
GENERATOR

102

OPERATION
CONTROL
UNIT

s LATCH r 106

SET LATCH

PK ALGORITHM

DES ALGORITHM

(CRYPTOGRAPHIC FACILITY)

(COMPUTER)

FIG 17

PARAMETER P4

OP3

101

TREG 108

102

OPERATION
CONTROL
UNIT

T

105

104

KT

D

REGISTER | D | E | P

C1

C2 = KF

s   LATCH   r ~106

PK ALGORITHM

DES ALGORITHM

(CRYPTOGRAPHIC FACILITY)

(COMPUTER)

FIG 18

PARAMETER P1

PARAMETER P2

101

OP4

KTTR5678

103

102

OPERATION
CONTROL
UNIT

E

104

NON-ZERO
CONSTANT

C1

D 105

REGISTER | D | E | P

C

C2

KF1

DES ALGORITHM

(CRYPTOGRAPHIC FACILITY)

(CCMPUTER)

FIG 19

(CRYPTOGRAPHIC FACILITY)

(COMPUTER)

FIG 20

(PASSWORD) P1

(PROG # | DISK #) P2

OP6

101

103

E

KTTR5678

102

OPERATION
CONTROL
UNIT

104

C1

D

105

REGISTER | D|E|P

C2 = KF

DES ALGORITHM

(CRYPTOGRAPHIC FACILITY)

(COMPUTER)

FIG 21

PARAMETER P4

OP7

101

TREG

108

KT

D

105

102

OPERATION
CONTROL
UNIT

T

C1

104

REGISTER | D|E|P

C2 = KF

s   LATCH   r

106

PK ALGORITHM

DES ALGORITHM

(CRYPTOGRAPHIC FACILITY)

(COMPUTER)